



GP2781

PATENT APPLICATION
Attorney Docket No. R/97005Q

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to:

Assistant Commissioner for Patents
Washington, D.C. 20231.

on March 29, 2000.

Typed or printed name of person signing this certificate

Clarke A. Wixon

Signature: 

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application of: Jean-Marc Andreoli et al.)

Appl. No.: 09/421,846)

Filed: October 20, 1999)

Art Unit: 2781

Examiner: Not Yet Assigned

Title: DOCUMENT CONSTRAINT DESCRIPTORS OBTAINED FROM USER SIGNALS
INDICATING ATTRIBUTE-VALUE RELATIONS

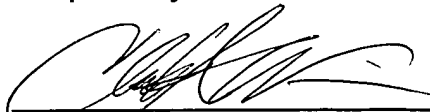
Assistant Commissioner for Patents
Washington, D.C. 20231

SUBMISSION OF PRIORITY DOCUMENTS UNDER 37 C.F.R. §1.55(a)

Sir:

In accordance with 37 C.F.R. §1.55(a), applicant submits herewith a certified copies of the foreign applications upon which they claim priority under 35 U.S.C. § 119.

Respectfully submitted,



Clarke A. Wixon
Attorney for Applicant(s)
Registration No. 39,878
Telephone: 650-812-4295

Xerox Corporation
Palo Alto, California
Date: March 29, 2000

RECEIVED
APR -4 2000
TECH CENT 2700



The
Patent
Office



INVESTOR IN PEOPLE

The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.

Signed

A handwritten signature in dark ink, appearing to be 'M. C. Jenkins'.

Dated 28-2-00

CERTIFIED COPY OF
PRIORITY DOCUMENT



Request for grant of a patent

(See the notes on the back of this form. You can also get an explanatory leaflet from the Patent Office to help you fill in this form)

The Patent Office

Cardiff Road
Newport

Gwent NP9 1RH

1. Your reference

R/97005Q1/JDR

2. Patent application number

(The Patent Office will fill in this part)

9708175.6

23 APR 1997

3. Full name, address and postcode of the or of each applicant (*underline all surnames*)

Xerox Corporation

Xerox Square
Rochester

New York 14644

United States of America

Patents ADP number (*if you know it*)

3996188001

If the applicant is a corporate body, give the country/state of its incorporation

New York, USA

4. Title of the invention

FEATURE CONSTRAINT BASED DOCUMENT INFORMATION RETRIEVAL AND DISTRIBUTION

5. Name of your agent (*if you have one*)

Julian D. Reynolds

"Address for service" in the United Kingdom to which all correspondence should be sent (*including the postcode*)Rank Xerox Limited
Patent Department
Parkway
Marlow
Bucks SL7 1YL
United KingdomPatents ADP number (*if you know it*)

6329718002

6. If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and (*if you know it*) the or each application number

Country

Priority application number
(*if you know it*)Date of filing
(*day / month / year*)

7. If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date of the earlier application

Number of earlier application

Date of filing
(*day / month / year*)8. Is a statement of inventorship and of right to grant of a patent required in support of this request? (*Answer 'Yes' if:*

Yes

- a) any applicant named in part 3 is not an inventor, or
- b) there is an inventor who is not named as an applicant, or

c) any named applicant is a corporate body.

See note (d))

9. Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document

Continuation sheets of this form

Description 12

Claim(s) 4

Abstract 1

Drawing(s) 9

10. If you are also filing any of the following, state how many against each item.

Priority documents

Translations of priority documents

Statement of inventorship and right to grant of a patent (*Patents Form 7/77*) 1

Request for preliminary examination and search (*Patents Form 9/77*) 1

Request for substantive examination (*Patents Form 10/77*)

Any other documents
(please specify)

11. I/We request the grant of a patent on the basis of this application.

Signature

Julian D. Reynolds
Julian D. Reynolds

Date 11.4.97

12. Name and daytime telephone number of person to contact in the United Kingdom Julian D. Reynolds (01628 893447)

Warning

After an application for a patent has been filed, the Comptroller of the Patent Office will consider whether publication or communication of the invention should be prohibited or restricted under Section 22 of the Patents Act 1977. You will be informed if it is necessary to prohibit or restrict your invention in this way. Furthermore, if you live in the United Kingdom, Section 23 of the Patents Act 1977 stops you from applying for a patent abroad without first getting written permission from the Patent Office unless an application has been filed at least 6 weeks beforehand in the United Kingdom for a patent for the same invention and either no direction prohibiting publication or communication has been given, or any such direction has been revoked.

Notes

- If you need help to fill in this form or you have any questions, please contact the Patent Office on 0645 500505.*
- Write your answers in capital letters using black ink or you may type them.*
- If there is not enough space for all the relevant details on any part of this form, please continue on a separate sheet of paper and write "see continuation sheet" in the relevant part(s). Any continuation sheet should be attached to this form.*
- If you have answered 'Yes' Patents Form 7/77 will need to be filed.*
- Once you have filled in the form you must remember to sign and date it.*
- For details of the fee and ways to pay please contact the Patent Office.*

1

**FEATURE CONSTRAINT BASED DOCUMENT
INFORMATION RETRIEVAL AND DISTRIBUTION**

The present invention relates to data processing, and more particularly relates to the transfer between computing devices, and the retrieval by such devices, of document related information. Even more particularly, the invention concerns such information retrieval and transfer using feature constraints.

While the use of portable computing devices is becoming more widespread, it is as yet not possible to store in such devices the electronic files for all the documents that a user may wish to have access to (e.g. due to storage capacity limitations), and the transfer of bulk documents between such devices, or between one such device and a desktop office or home computer may be time consuming or otherwise user unfriendly. At the same time, there is growth in the provision of electronic document repositories and expanding use of document transfer via the internet.

This situation is addressed in EP-A-691,619 (hereafter "EP'619"), which discloses a system including any number workstations, file servers, printers and other fixed devices (including multifunction devices) coupled in a network, and a number of portable devices (e.g. handheld or wristwatch computer) carried by users and coupled to the network by infrared (IR) link. Each portable device emulates its user's personal satchel for documents: the device is programmed to receive transmit and store document references (World Wide Web URLs), each of which is associated with an electronic document stored in an electronic repository at a site on the web. Documents are distributed from one person to another by transmission of document URLs, and a document is sent to a printer by beaming that document's URL to an IR transceiver associated with that printer.

A problem with such a system is that it stores and distributes links (Web URLs) only to single documents: each document reference points to a particular document. A further problem is how to keep track of a continually-changing electronic document repository, to dynamically adjust to changing document descriptions (updated author list, updates dates, updated keywords, etc.) For example, in the device of EP'619 the user may be carrying around a URL for a document A written by an author B on the subject of topic C. Unknown to the user, there may be in existence author B's updated document A', written 2 years after A, on the same topic C, or there may be a further document A" by the same author and on a topic closely related to C (and, for example, having a keyword in common). The user does not have the link to A' or A" and, even though they would be of use to him, without carrying out a further searching/browsing exercise to look for them and retrieve the URLs, he would continue about his work oblivious of them.

It would be desirable to have a system which avoided the lack of dynamicity of existing systems, and which enabled users to have to hand pointers to a *set* of documents which may be dynamically changing.

The present invention provides a method carried out in a data processing device including a memory, and a user interface, comprising: (a) receiving at least one user input, the user input defining at least one relation, the or each relation defining a document related entity and a

property of the entity, (b) compiling a feature constraint, the feature constraint including the relation(s) received in step (a).

The present invention further provides a data processing device when suitably programmed for carrying out the method of any of the appended claims, the device comprising a processor, a memory, and a user interface.

The present invention further provides a data processing device, system or apparatus according to claims 19-22 of the appended claims.

The present invention further provides a data processing device comprising: a processor, a memory coupled to the processor, and a user interface coupled to the processor and to the memory and adapted to be operable by a user to generate user inputs, means for receiving at least one user input, the user input(s) defining at least one relation, the or each relation defining a document related entity and a property of the entity, and for compiling a feature constraint, the feature constraint including the received relations.

The aforementioned problems are overcome by using a constraint description: that is, instead of using a pointer to a single document or to a static set of documents, feature constraints relating to documents are stored and distributed, the feature constraints providing pointers to a dynamic set of documents satisfying the constraints given.

An advantage is apparent from a consideration of an example. Supposing the feature constraint constrains only the author name (e.g. Smith) and the date of the document (e.g. >1995), then this will point to any document authored by Smith later than 1995. If, then, in 1998, Smith authors another document, this document is automatically linked with the 'old' constraint, without further user intervention.

A further benefit is in the transfer of document references between the portable devices of multiple users when the users (for example in a meeting at a remote location) are away from their office workstations. If the sender knows of there is a document in existence that he'd like to send, but doesn't have the URL on his device, or cannot recall how to identify it on his device and therefore cannot send the URL, but he knows an author's name, a date (range) of creation, and/or a keyword, he can thus send a feature constraint to the recipient. The recipient can then use the feature constraint to retrieve the document on his desktop workstation once he returns to his office.

Embodiments of the invention will now be described, by way of example, with reference to the accompanying drawings, in which:

Figure 1 illustrates schematically a data processing network which may be used to implement an embodiment of the invention;

Figure 2 shows a portable computing device used in accordance with an embodiment of the invention;

Figure 3 illustrates schematically the scope defined by a signed feature constraint;

Figure 4 is a schematic flow chart of the steps in compiling a feature constraint;

Figure 5 is a schematic diagram of the user interface of a fixed computing device at one instant during the entry by a user of a query;

Figure 6 is a schematic diagram of the user interface of a portable computing device at one instant;

Figure 7 is a schematic flow chart of the steps in sending a feature constraint from one (portable/fixed) device to another (portable/fixed) device;

Figure 8 is a schematic illustration of a data packet encoding a feature constraint;

Figure 9 is a schematic flow chart of the steps in using a feature constraint to retrieve document references and display or print corresponding documents

Figure 10 illustrates a portion of a list of hits obtained during the process of Fig. 9;

Figure 11 shows selected hits from the list of Fig. 10 after transformation into HTML format; and

Figure 12 illustrates a more detailed presentation of a single selected hit.

1. System hardware

It will be appreciated that the present invention may be implemented using conventional computer network technology, either using a local area network (LAN) or, more suitably, a wide area network (WAN). The invention has been implemented using conventional web browser software (e.g. Netscape) providing cross-platform communication and document transfer over the internet. This is schematically illustrated in Fig. 1. It will be appreciated that each machine 22, 24, 26 forming part of the network 21 may be a PC running Windows™, a Mac running MacOS, or a minicomputer running UNIX, which are well known in the art. For example, the PC hardware configuration is discussed in detail in *The Art of Electronics*, 2nd Edn, Ch. 10, P. Horowitz and W. Hill, Cambridge University Press, 1989. However, it will be appreciated that the invention may be implemented using different system configurations: see EP'619.

Upon request of a user at machine 22 a document stored on machine 26 may be retrieved and sent from machine 26 over the internet, via any number of intermediate machine 24 to machine 22. As is well known, the document may be retrieved using as a unique identifier its World Wide Web URL, as discussed in EP'619. Preferably also connected to the network 21 are any number of printers or multifunction devices (capable of scanning/printing/faxing, etc.) (not shown), as discussed in EP'619. Multifunction devices are discussed in more detail in EP-A-741,487. Each machine coupled to the network is also equipped with appropriate hardware and software, which is known in the art, for communication with portable computing devices, such as personal digital assistants (PDAs), handheld PCs, or pocket or wristwatch computers.

A preferred form of portable computing device is shown in Fig. 2. This device 2 is discussed in more detail in EP'619. However, it will be appreciated that the computing device may be fabricated in a multitude of forms: for example, the device 2 may be of any of the forms disclosed in European patent application EP-A-_____ (attorney's reference D/95594/JDR), corresponding to U.S. application S.N. 08/_____ ("Personal storage device for application and data transfer", filed 28 March 1996). In its preferred form, computing device 2, and each of the fixed machines on the network 21, are equipped for infrared communication and suitably, the data packets transmitted between the computing device 2 and

the fixed machines, and enabling the document(s) to be retrieved, conform to the physical and link layer formats (IrLAP) described in the industry standard Infrared Data Association (IrDA) specification, version 1.0, which is well known in the art.

2. Knowledge brokers

According to one embodiment of the invention, in retrieving a document for local or remote printing, or transfer to another machine, each machine in Fig. 1 employs brokers.

Brokers are software agents which can process knowledge search requests. Knowledge is taken here to be any piece of electronic information intended to be publicly accessible. Different, possibly distributed, information sources are assumed to be available, from a simple file in a user's directory to a database local to a site, up to a wide area information service (WAIS) on the internet, for example.

When receiving a request, a broker may have sufficient knowledge to process it, or may need to retrieve more knowledge. For that purpose, it releases sub-requests, aimed at other brokers. Thus, knowledge retrieval is achieved by the collaboration of all the brokers which are alternatively service providers processing requests and clients of these services generating sub-requests. The infra-structure required to support such collaboration, and the way knowledge is stored locally within each broker is not discussed in detail here (for details of a suitable model, the reader is referred to Andreoli et al (1996), *The Constraint-Based Knowledge Broker Model: Semantics, Implementation and Analysis, J. Symbolic Computation*). The following discussion addresses rather the knowledge manipulations occurring within each broker.

In order to collaborate, the brokers must at least understand each other. This means that all the requests must be formulated in a common language (and also all the answers to the requests), even if the brokers may perform local translations. Logic provides the adequate language for such a purpose. A request can be expressed by a pair $\langle x, P \rangle$ where x is a logical variable and P a logical formula involving x , meaning "*Retrieve knowledge objects x such that the property expressed by formula P holds*". Interestingly, an answer to such a request can be expressed in the same formalism, i.e. a pair $\langle x, Q \rangle$ meaning "*There exists a knowledge object x satisfying the property expressed by formula Q* ". The requirement here is that P must be a logical consequence of Q , so that the answer contains at least as much knowledge as the request. Moreover, the same logical formalism can be used to capture the scope of a broker, i.e. the area of knowledge it is concerned with: a broker with scope $\langle x, R \rangle$ means "*I am not capable of retrieving knowledge objects x which do not satisfy the property expressed by formula R* ". In many situations, the scope of a broker may vary, because it gets specialised or, on the contrary, expands its capacities, either externally or due to the knowledge retrieval process itself.

In other words, logic provides a common language where both requests, answers and scopes can be expressed. Brokers then perform logical operations on these three components. The most important logical operation, from which all the others can be reconstructed, is satisfiability checking, i.e. deciding whether some object could satisfy the property expressed by a formula, or, on the contrary, whether it is intrinsically contradictory. Unfortunately, it is well known that this operation, for *full* Classical Logic, is not algorithmic, i.e. it is provably impossible to write a program which implements it and always

terminates. Given this limitation, a lot of research in knowledge representation has been focused on identifying *fragments* of Classical Logic in which satisfiability is algorithmically decidable. The trade-off here is between expressive power and tractability: the empty fragment, for example, is obviously tractable, but it is not very expressive.

The most popular fragment which emerged is known as "feature constraints" (FCs). The satisfiability problem in this case is also known as "feature constraint solving".

As is known, feature constraints are built from atomic constraints which are either sorts or features. A sort is a unary relation, expressing a property of a single entity. For example, $P:person$ expresses that an entity P is of sort *person*. A feature is a binary relation expressing a property linking two entities. For example, $P:employer \rightarrow E$ expresses that entity P has an employer, which is an entity E . Apart from sorts and features, most feature systems also allow built-in relations such as equality and disequality.

3. Principles of feature constraints

The full fragment of feature constraints, where the atomic components mentioned above are allowed to be combined by all the logical connectives (conjunction, disjunction, negation and quantifiers), although very expressive, is hardly tractable. A subfragment called "basic feature constraints" (BFC) has been considered, where negation and disjunction are simply forbidden. Efficient constraint solving algorithms have been proposed for this sub-fragment. However, a drawback is that the complete absence of negation puts strong limitations on the kind of operations a knowledge broker may wish to perform.

In preferred embodiments, the present invention makes use of a powerful operation, referred to as "scope-splitting", which relies on the use of negation. Indeed, a broker may wish to split its scope, specified by a pair $\langle x, P \rangle$ according to a criterion expressed by a formula F , thus creating two brokers with scope $P \wedge F$ and $P \wedge \neg F$. Thus, a broker in charge of bibliographic information may wish to split its scope into two new scopes: "books written after 1950", which can be represented by the BFC

x

$x : book$

$x : year \rightarrow y$

$y > 1950$

and its complement, i.e. "books written before 1950 or documents which are *not* books"; this latter scope cannot be expressed using BFC, because negation and disjunction cannot be dispensed with. It has been discovered that the scope splitting operation is useful in many situations, for example to implement brokers capable of memorising and re-using information gathered during their lifetime. Embodiments of the present invention make use of, on the one hand, a fragment of feature constraints, called "signed feature constraints" (SFC), which allows limited use of negation, precisely capable of expressing the kind of split scope mentioned above, and on the other hand, an efficient constraint solving method for SFC.

3.1 Signed Feature Constraints

A signed feature constraint is composed of a positive part and a list of negative parts, both of them being basic feature constraints. For example, the following signed feature constraint

```
P
+ P : person,
  P : employer-> E,
  E : "Xerox"
- P : nationality-> N,
  N : "American"
- P : spouse-> P'
  P' : person
  P' : employer-> E'
  E' : "Xerox"
```

specifies a Xerox employee who is not American and is not married to another Xerox employee. This is represent it graphically as in Fig.3. The round boxes denote the entities (logical variables), the sort relations (unary) are represented by dashed arrows labelled by the name of the sort in a square box, the feature relations (binary) are represented by plain arrows labelled by the name of the feature in a square box. The built-in predicates (not present in the example) are represented by rhombuses. The positive part of the SFC is contained in the top box and marks the distinguished entity of the scope (P in the example) by a double round box. The negative parts of the SFC are contained in the lower boxes in grey.

The main interest of SFC comes from the following property:

If the scope of a broker is represented by an SFC e_0 , and this scope is split by a BFC e , then the two resulting split scopes e^+ , e^- are both SFC.

Indeed, e^+ is obtained by merging the positive part of e_0 with the BFC e ; and e^- is obtained by extending e_0 with a new negative part containing e alone. For example, assume a broker in charge of a bibliographical database containing various documents (books, videos etc.) about Art, but not authored by an American. It is represented by the SFC

```
X
+X : topic-> T
  T : "Art"
-X : author-> A
  A : nationality-> N
  N : "American"
```

It may be split by the constraint "books written after 1950", expressed by the BFC

```
X
X : book
X : year-> Y
Y > 1950
```

The resulting scopes are simply

X

+X : book

X : topic-> T

X : year-> Y

T : "Art"

Y > 1950

-X : author-> A

A : nationality-> N

N : "American"

i.e. "Art books written after 1950 but not by an American author" and

X

+X : topic-> T

T : "Art"

-X : author-> A

A : nationality-> N

N : "American"

-X : book

X : year-> Y

Y > 1950

i.e. "Art documents not authored by an American but not books subsequent to 1950".

3.2 Solving Signed Feature Constraints

Most constraint systems make a number of assumptions on the nature of sorts and features, called the axioms of the systems. These axioms are crucial to the satisfiability algorithm, since they determine when a feature constraint is contradictory and when it is satisfiable.

3.2.1 Feature Axioms

For the purpose of simplicity, the embodiment disclosed here makes use of a slight variant of the basic axiom system used in Ait-Kaci H. *et al.* (1994), A Feature-Based Constraint-System for Logic Programming with Entailment, *Theoretical Computer Science* 122, pp. 263–283, although it will be appreciated by persons skilled in the art that the principles of the method apply to other sets of axioms as well.

1. Features are functional: this means that if two pairs of entities which are constrained by the same feature have the same first term, they also have the same second term. For example, it can be considered that the feature *spouse* is functional (within a specific cultural setting), meaning that a person cannot have two spouses: if, for a person X, we have $X:\text{spouse} \rightarrow Y$ and $X:\text{spouse} \rightarrow Z$, then the entities Y and Z coincide (i.e. denote the same person). Other systems allow multi-valued features.

2. Sorts are disjoint: this means that no entity can be of two distinct sorts. For example, a book is not a person: we cannot have an entity x with $x:\text{book}$ and $x:\text{person}$. Other systems consider hierarchies of sorts where some entities can have multiple sorts as long as they have a common denominator in the hierarchy.

3. There is a distinguished subset of sorts, called "value" sorts, so that no two distinct entities can be of the same value sort. Traditional basic elements (strings, numbers, etc.) are typical value sorts: for example, the string "Xerox" or the number 1950 are value sorts. Value sorts are not allowed to have features: this is the only axiom connecting sorts and features. Other systems consider more refined connections between sorts and features.

4. There is a distinguished built-in binary predicate, equality, with the traditional congruence axioms (which involve sorts and features). The axioms describing all the other built-in predicates are assumed to contain no mention of sorts and features.

These axioms are formally written in section A: Axioms in the Appendix at the end of this disclosure. They form a theory T .

3.2.2 Constraint Satisfaction

First, it is assumed that satisfiability over built-in predicates is decidable. This means that there is an algorithm which, given a formula F using only built-in predicates (F is also called a built-in constraint), can decide whether F is a logical consequence of the theory T (written $\vdash_T F$).

Constraint satisfaction over BFCs is defined by a set of conditional rewrite rules over BFCs (section B.1 of the Appendix) which have the following properties

(a) *The system of rules is convergent and hence defines a "normal form" for BFCs.* This can be shown in a classical way by proving that the system is "Church-Rosser" (critical pairs converge) and "Noetherian" (the size of the terms strictly decrease by rewriting).

(b) *A BFC is satisfiable if and only if its normal form is not reduced to the contradiction.* One implication can be proved by showing that rewrite steps preserve satisfiability. The reverse implication can be proved by displaying a model which satisfies BFCs whose normal form is not reduced to the contradiction.

Thus the rewrite rules describe the steps of the constraint satisfaction algorithm. This algorithm always terminates because the system of rewrite rules is convergent. It is to be noted that the definition of the rules rely on satisfiability tests of built-in constraints, which has been assumed decidable. This means that the algorithm is modular and can accommodate any kind of built-in constraints as long as a proper built-in constraint satisfaction algorithm is provided.

Using rewrite rules for constraint satisfaction algorithm can be implemented in a naive way in some symbolic language like Lisp or Prolog, or can be optimised, taking into account the properties of the specific built-in constraints which are used.

The algorithm for constraint satisfaction over SFCs (section B.2 of the Appendix) can informally be described as follows. Given an SFC, its positive component is first normalised by the algorithm for BFCs. If the result is a contradiction, the whole SFC is unsatisfiable. Otherwise, the positive

component (normalised) is inserted in each of the negative components, which are then normalised by the algorithm for BFCs. If a resulting negative component has a contradictory normal form, it is eliminated, and if it has a tautological normal form the whole SFC is unsatisfiable. The normal form for SFCs thus obtained has the following property:

An SFC is satisfiable if and only if its normal form is not reduced to the contradiction. As in the previous case, the difficult part of the implication can be proved using model theory.

4. User transactions with feature constraints

Figure 4 is a schematic flow chart of the steps in compiling a feature constraint. Initially, (s41) the multiple user inputs are received as the user specifies a query, e.g. "books or articles after 1990 in which the title contains "constraints" but does not contain "internet". As the inputs are received, relations (see section 2 above) are determined. Figure 5 is a view of the user interface of a fixed computing device at one instant during the entry by a user of a query: this is discussed further in British patent application 97 _____ (agent's ref. R/97005Q2/JDR), filed contemporaneously herewith. Then, once the user has finished inputting the elements of the query, the feature constraint is compiled (step s42) from the relations obtained. The user may be prompted (s43) to enter a short query name, or a default one may be assigned, and the FC stored in its final form.

This inputting by a user may be performed on a portable computing device in a manner known in the art, using a keyboard, touchscreen user interface, displayed lists and scrolling/control buttons etc. If the user interface for the portable device is small, the method disclosed in EP-A-733,964 — for accessing members of a stored data set — may be used. A further alternative is for the user to input the query, from which the FC is obtained, on a piece of paper via a scanner: the user checks boxes on a pre-printed standard sheet (on which possible relations are already indicated in human-readable form); and when the completed sheet is scanned, the relations are extracted from the resulting image data and used to compile the FC. This alternative is discussed further in the aforementioned GB appln. (agent's ref. R/97005Q2/JDR).

Figure 6 is a schematic diagram of the user interface 60 of a portable computing device at one instant. The general techniques disclosed in EP'619 may be used in presenting information to the user. As in EP'619, icons 61 and document names 62 may be displayed which correspond to stored URLs. However, there is also displayed an icon 63 and short query name 64 corresponding to a stored FC. Optionally, the user may view the long version, for example by selecting icon 63 and performing button presses corresponding to a double mouse click in a conventional user interface. In this case, this might display the long version 'Books by Babbage after 1993 with title including "lecture" and "Object-Oriented" '.

Figure 7 is a schematic flow chart of the steps in sending a feature constraint from one (portable/fixed) device to another (portable/fixed) device. As described in EP'619, the computing devices are context sensitive: an icon for each other user's device in range of IR communication is displayed on the user interface (step s71). After the user has requested sending a FC to another user in range (s72), if the request concerns a new, as yet non-existent FC, then the process of Fig. 4 is followed, otherwise the

FC is retrieved from memory (s73). Then the FC is encoded in appropriate form and broadcast using any of the communications methods mentioned in EP'619. A data packet in which the FC may be encoded in the data field is schematically illustrated in Fig 8. However, the preferred means of communication is IR signals, and in particular by means of IrDA standard data transfer.

Returning to Fig. 7, the recipient's user interface also displays icons for all in-range users (step s75). On receiving an IR data packet, this is decoded to extract the FC (s77) and then stored in the designated location in the recipient device directory for future use, if required (s78).

Figure 9 is a schematic flow chart of the steps in using a feature constraint to retrieve document references and display or print corresponding documents. This may be performed by a conventional computer device, or by a multifunction device or printer equipped with a user interface. Initially, a FC is received from a user, for example in a data packet from a user of a portable device, as illustrated in Fig. 8, or by input directly into the machine by a user operating a keyboard and mouse, or touch screen, as is well known in the art. Upon the instigation of the user by appropriate input, the FC is solved as described in section 3 above, and the resulting request in the appropriate form passed to the search engine (s92). The search request is used to search all available repositories for documents satisfying the FC (s93); and if necessary, the request may be broken down into subrequests as discussed in more detail in the aforementioned British patent application (agent's reference R/97005Q2).

Once obtained by the search engine, a list of hits — of documents satisfying the FC — is displayed (s94), as shown in Fig. 10. Then, in response to appropriate user input, operations may be performed to display individual hits with expanded detail of the document, to convert the document information to HTML format, or to download the document from the repository (s95).

Figure 11 shows selected hits from the list of Fig. 10 after transformation into HTML format. As can be seen, for each hit there is displayed further information, such as author name, http_url, information source and title. If desired, the user can view the document for hit 1 by mouse clicking on the http_url displayed. The document can then be printed, if needed (s96).

Figure 12 illustrates a more detailed presentation of a single selected hit, i.e. with a set of attributes of the document. It can be seen that against one or more of the attributes are displayed URLs providing links to further pages providing information related to those attributes.

It will be appreciated that in implementing the above described techniques, feature constraints in the form of signed feature constraints, as discussed in section 3, may be employed.

Appendix

A Axioms of the System

There are three sets of axioms.

Specific axioms for features and sorts :

Let τ, τ' denote any sorts, and f denote any feature.

$$\begin{aligned} \forall x, y, z \quad x \xrightarrow{f} y \wedge x \xrightarrow{f} z \supset y = z \\ \forall x \quad \neg(x : \tau \wedge x : \tau') \text{ if } \tau \neq \tau' \\ \forall x, y \quad x : \tau \wedge y : \tau \supset x = y \text{ if } \tau \text{ is a value sort} \\ \forall x, y \quad \neg(x : \tau \wedge x \xrightarrow{f} y) \text{ if } \tau \text{ is a value sort} \end{aligned}$$

Congruence axioms for equality :

Let p denote any built-in predicate. The traditional congruence axioms are:

$$\begin{aligned} \forall x \quad x = x \\ \forall x, y \quad x = y \supset y = x \\ \forall x, y, z \quad x = y \wedge y = z \supset x = z \\ \forall x, y \quad x : \tau \wedge x = y \supset y : \tau \\ \forall x, y, z \quad x \xrightarrow{f} y \wedge x = z \supset z \xrightarrow{f} y \\ \forall x, y, z \quad x \xrightarrow{f} y \wedge y = z \supset x \xrightarrow{f} z \\ \forall \bar{x}, y \quad p(\bar{x}) \wedge x_i = y \supset p(\bar{y}) \end{aligned}$$

where i is some index in the list of variable \bar{x} and \bar{y} is identical to \bar{x} except that $y_i = y$.

Built-in predicate axioms :

They must not mention sorts and features. For example, disequality can be axiomatized by

$$\begin{aligned} \forall x, y \quad x \neq y \vee x = y \\ \forall x \quad \neg(x \neq x) \end{aligned}$$

Precedence constraints are axiomatized by

$$\begin{aligned} \forall x \quad \neg(x < x) \\ \forall x, y, z \quad x < y \wedge y < z \supset x < z \end{aligned}$$

The built-in predicates $>, \leq, \geq$ can then be defined from $<$ and equality.

B Constraint Satisfaction

B.1 The BFC case

We represent a BFC as a pair $\langle B \mid \Gamma \rangle$ where B is a built-in constraint and Γ an *unordered* list of sort and feature constraints (read conjunctively). \perp denotes the contradiction.

There are two sets of rewrite rules. The following rules correspond to simplifications of the BFCs.

$$\begin{aligned}
\langle B \mid x \xrightarrow{f} y, z \xrightarrow{f} t, \Gamma \rangle &\mapsto \langle B \wedge y = t \mid x \xrightarrow{f} y, \Gamma \rangle \text{ if } \vdash_T B \supset x = z \\
\langle B \mid x : \tau, y : \tau, \Gamma \rangle &\mapsto \langle B \mid x : \tau, \Gamma \rangle \text{ if } \vdash_T B \supset x = y \text{ and } \tau \text{ is not a value sort} \\
\langle B \mid x : \tau, y : \tau, \Gamma \rangle &\mapsto \langle B \wedge x = y \mid x : \tau, \Gamma \rangle \text{ if } \tau \text{ is a value sort}
\end{aligned}$$

The following rules correspond to the detection of inconsistencies.

$$\begin{aligned}
\langle B \mid \Gamma \rangle &\mapsto \perp \text{ if } \vdash_T \neg B \\
\langle B \mid x : \tau, y : \tau', \Gamma \rangle &\mapsto \perp \text{ if } \vdash_T B \supset x = y \text{ and } \tau \neq \tau' \\
\langle B \mid x : \tau, y \xrightarrow{f} z, \Gamma \rangle &\mapsto \perp \text{ if } \vdash_T B \supset x = y \text{ and } \tau \text{ is a value sort}
\end{aligned}$$

The following property justifies the algorithm

$$\langle B \mid \Gamma \rangle \mapsto \perp \text{ if and only if } \vdash_T \forall \neg (B \wedge \bigwedge_{c \in \Gamma} c)$$

B.2 The SFC case

We represent an SFC as an *unordered* list of BFCs prefixed with a sign (+ or -); by definition, one and only one component is positive. Let S be an SFC. The SFC-normal form of S is written S^* and is obtained by the following algorithm:

Let c_o be the BFC-normal form of the positive component of S .
If $c_o = \perp$ **Then**
 Return \perp
Else
 c_o is of the form $\langle B_o \mid \Gamma_o \rangle$
 Let $\{\langle B_i \mid \Gamma_i \rangle\}_{i=1, \dots, n}$ be the list of negative components of S .
 Foreach $i = 1, \dots, n$
 Let c_i be the BFC normal form of $\langle B_o \wedge B_i \mid \Gamma_o, \Gamma_i \rangle$.
 If there exists $i \in 1, \dots, n$ such that $c_i = \langle B \mid \Gamma \rangle$ and $\vdash_T B$ and Γ is empty **Then**
 Return \perp
 Else
 Let $I = \{i \in 1, \dots, n \text{ such that } c_i \neq \perp\}$
 Return $\{+c_o, \{-c_i\}_{i \in I}\}$

The following property justifies the algorithm

$$[+\langle B_o \mid \Gamma_o \rangle, \{-\langle B_i \mid \Gamma_i \rangle\}_{i=1}^n]^* = \perp \text{ if and only if } \vdash_T \forall \neg [(B_o \wedge \bigwedge_{c \in \Gamma_o} c) \wedge \bigwedge_{i=1}^n \neg (B_i \wedge \bigwedge_{c \in \Gamma_i} c)]$$

Claims:

1. A method carried out in a data processing device including a processor, memory, and a user interface, comprising:

(a) receiving at least one user input, the user input(s) defining at least one relation, the or each relation defining a document related entity and a property of the entity,

(b) compiling a feature constraint, the feature constraint including the relation(s) received in step (a).

2. The method of claim 1, wherein, prior to step (b), step (a) is repeated a plurality of times.

3. The method of claim 1 or 2, wherein the data processing device comprising device comprises a portable computing device, and step (a) comprises receiving user inputs via touchscreen or a keyboard.

4. The method of claim 1 or 2, wherein the data processing device comprising device comprises a fixed computing device, and step (a) comprises receiving user inputs via touchscreen, a keyboard, and/or mouse.

5. The method of claim 4, wherein the fixed computing device comprises a multifunction device including scanning means,

wherein step a) comprises:

(a1) scanning an image-bearing portable medium to produce electronic signals corresponding to at least one predetermined portion of the medium; and

(a2) determining from said signals a relation corresponding thereto.

6. The method of claim 4, wherein said medium bears, for each said predetermined portion, a human readable indication of the disposition of said predetermined portion and, associated therewith, a human readable designator of the relation corresponding to said predetermined portion.

7. A method carried out in a data processing device including a processor, memory, and a user interface, comprising:

(d) receiving a first user input designating a graphical object corresponding to a stored feature constraint, the feature constraint comprising one or more relations, the or each relation defining a document related entity and a property of the entity,

(e) receiving a second user input indicating that the feature constraint is to be sent to another data processing device,

(f) encoding the feature constraint in a data packet, and

(g) transmitting the data packet.

8. A method carried out in a data processing device including a processor, memory, and a user interface, comprising:

- (h) receiving a data packet
- (i) decoding the data packet to derive a feature constraint, the feature constraint comprising one or more relations, the or each relation defining a document related entity and a property of the entity,
- (j) storing the feature constraint derived in step (i).

9. A method carried out in a data processing device including a processor, memory, and a user interface, the data processing device being couple in a network to one or more other data processing devices, at least one of the data processing devices including means for storing a repository of electronic documents, comprising:

- (k) receiving at least one user input designating a feature constraint, the feature constraint comprising one or more relations, the or each relation defining a document related entity and a property of the entity,
- (l) solving the feature constraint to determine therefrom one or more document references, the or each document reference corresponding to a document within said repository satisfying said feature constraint.

10. The method of claim, wherein step (k) comprises the method of claim 8.

11. The method of claim 9, further comprising the step of:

- (m) displaying the or each document reference determined in step (l).

12. The method of claim 11, further comprising the step of:

- (n) in response to a second user input designating one of the displayed document references, retrieving the document corresponding to said document reference from said repository and, optionally, displaying said document or a portion thereof.

13. The method of claim 11 or 12, further comprising the step of:

- (o) in response to a third user input, causing the document corresponding to said document reference to be printed.

14. The method of any of the preceding claims, wherein said feature constraint comprises at least a positive component and a negative component, each of the positive component and the negative component including one or more relations.

15. The method of any of the preceding claims, wherein one of said relations is a sort relation, the sort relation indicating the sort of which the respective entity comprises.

16. The method of any of the preceding claims, wherein one of said relations is a feature relation, the feature relation indicating that the respective entity has an attribute, the value of said attribute being a second entity.

17. The method of any of the preceding claims, wherein the property is a value, or range of values, which said entity may have.

18. A data processing device when suitably programmed for carrying out the method of any of the preceding claims, the device comprising a processor, a memory, and a user interface.

19. A data processing device comprising:
a processor,
a memory coupled to the processor, and
a user interface coupled to the processor and to the memory and adapted to be operable by a user to generate user inputs,
means for receiving at least one user input, the user input(s) defining at least one relation, the or each relation defining a document related entity and a property of the entity, and for compiling a feature constraint, the feature constraint including the received relations.

20. A system for accessing or distributing electronic documents, including: a repository of electronic documents, each document having a corresponding document reference, and a plurality of objects, at least one of said objects being portable or mobile, each object including means for communicating with the or each other object and with a user interface, and means for receiving, storing and/or transmitting a feature constraint, the feature constraint comprising one or more relations, the or each relation defining a document related entity and a property of the entity.

21. A portable device for accessing or distributing electronic documents, including: means for communicating with fixed or mobile electronic devices and with a user interface, at least one of said devices including means storing a repository of electronic documents, each document having a corresponding document reference, and means for receiving, storing and/or transmitting a feature constraint, the feature constraint comprising one or more relations, the or each relation defining a document related entity and a property of the entity,

22. An apparatus for scanning, copying and/or printing documents, including: means for accessing a repository of electronic documents, each electronic document having a corresponding document reference, means for communicating with one or more of a plurality of objects, at least one of said objects being portable or mobile, and with a user interface, and means for receiving, storing and/or

transmitting a feature constraint, the feature constraint comprising one or more relations, the or each relation defining a document related entity and a property of the entity,

ABSTRACT

FEATURE CONSTRAINT BASED DOCUMENT INFORMATION RETRIEVAL AND DISTRIBUTION

A system including any number workstations, file servers, printers and other fixed devices coupled in a network, and a number of portable devices carried by users and coupled to the network by infrared (IR) link. Each portable device emulates its user's personal satchel for documents: the device is programmed to receive transmit and store feature constraints (FCs). The FCs are distributed from one person to another by transmission of IR data packets, and once a network-connected computer, printer or multifunction device has received a FC via IR transceiver associated with it, a search request can be derived from the FC and document references (e.g. WWW URLs) satisfying the FC can be obtained via a search engine. For each URL obtained, the corresponding electronic document can be retrieved from a repository and displayed or printed.. The portable device is preferably a handheld or wristwatch computer with a graphical display for enabling the user to transfer FCs, and the fixed devices preferably include a scanner/copier/printer having its own IR transceiver.

[Fig. 3]

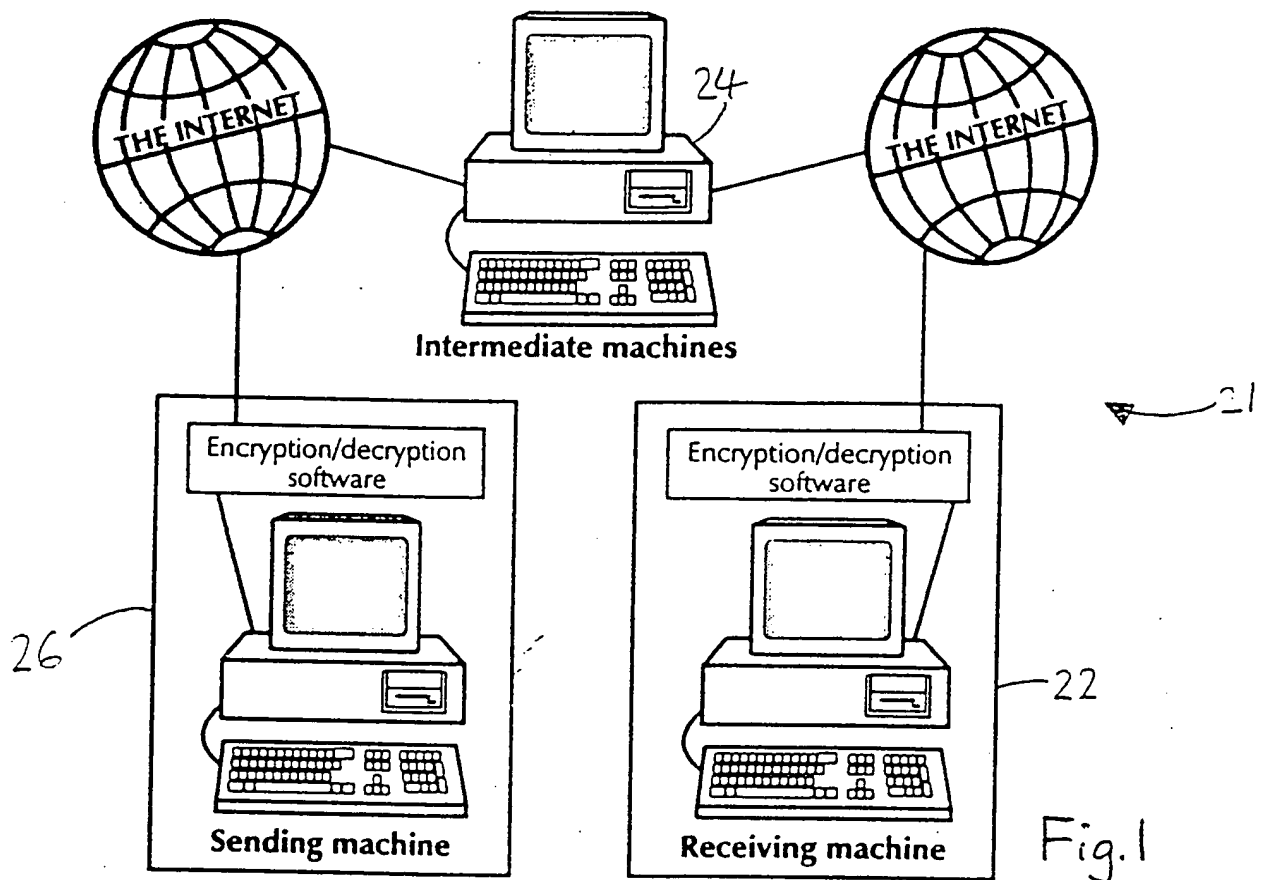
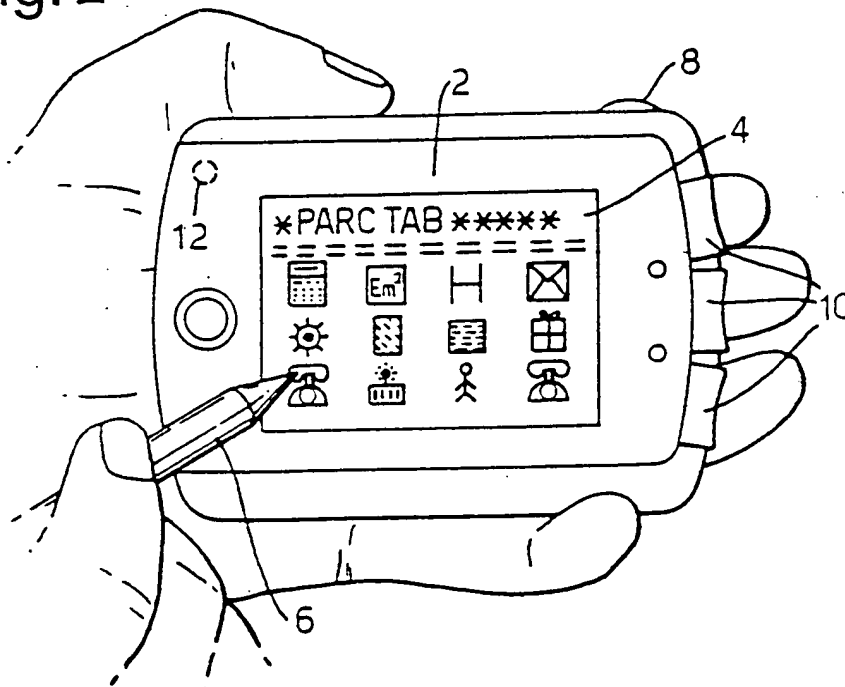


Fig. 1

Fig. 2



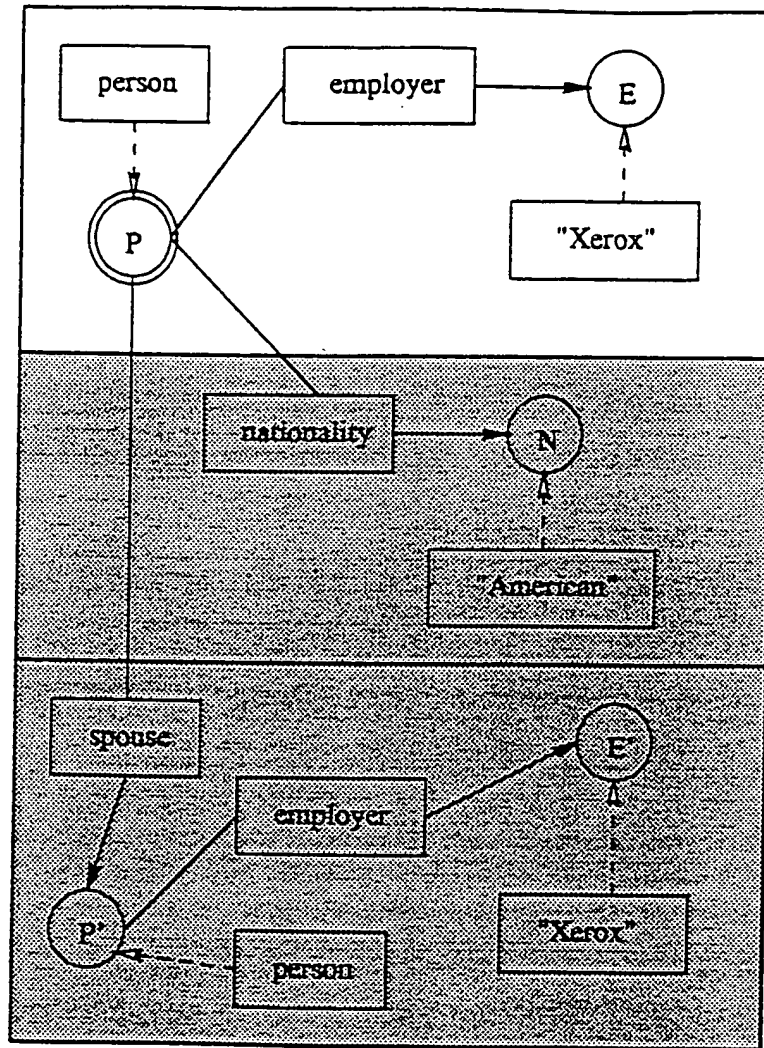
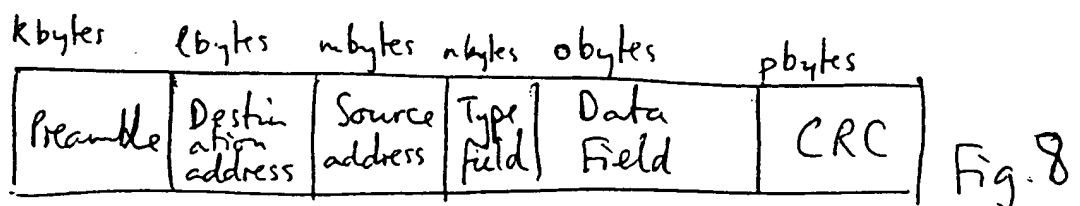
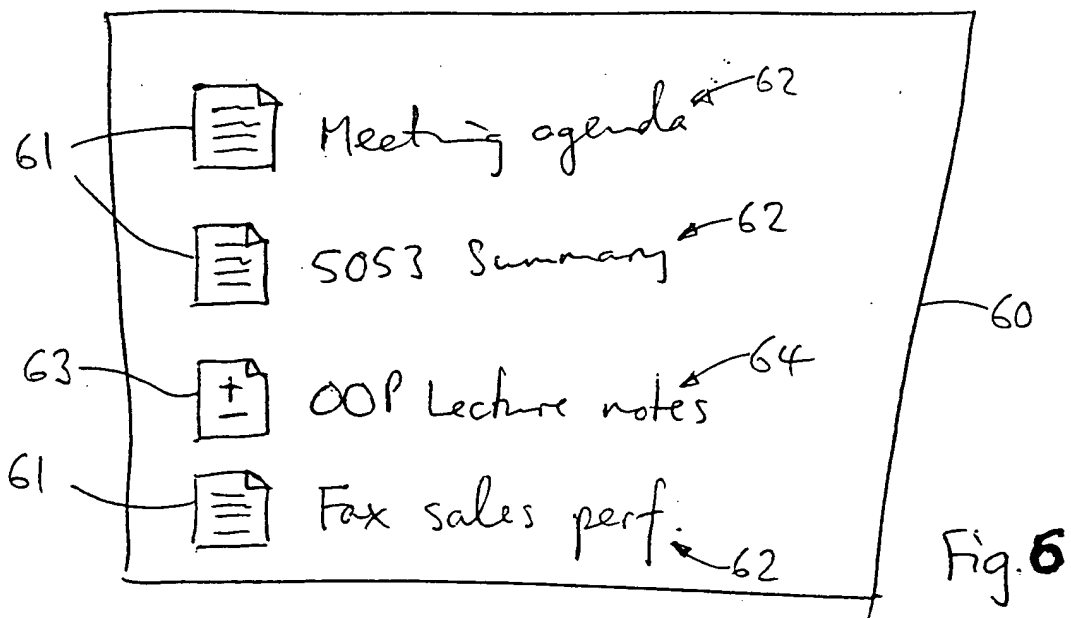
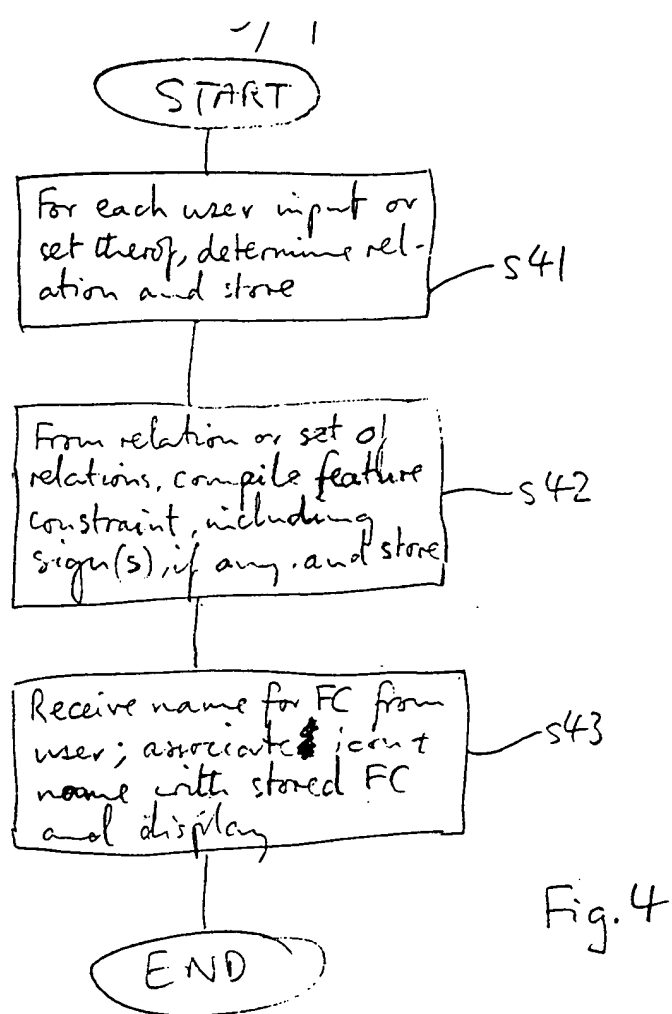


Fig.3



Netscape: RXRC Constraint-Based Knowledge Brokers

File Edit View Go Bookmarks Options Directory Window Help

Back Forward Home Reload Images Open Print Find Stop

Location:

What's New What's Cool Handbook Net Search Net Directory Software

Constraint-Based Knowledge Brokers

mainquery

Query concerning

title ☐ contains not ☐

Current Specification:

Global Options:

☐ Casesensitive Search

☐ Search also AltaVista

Fig. 5

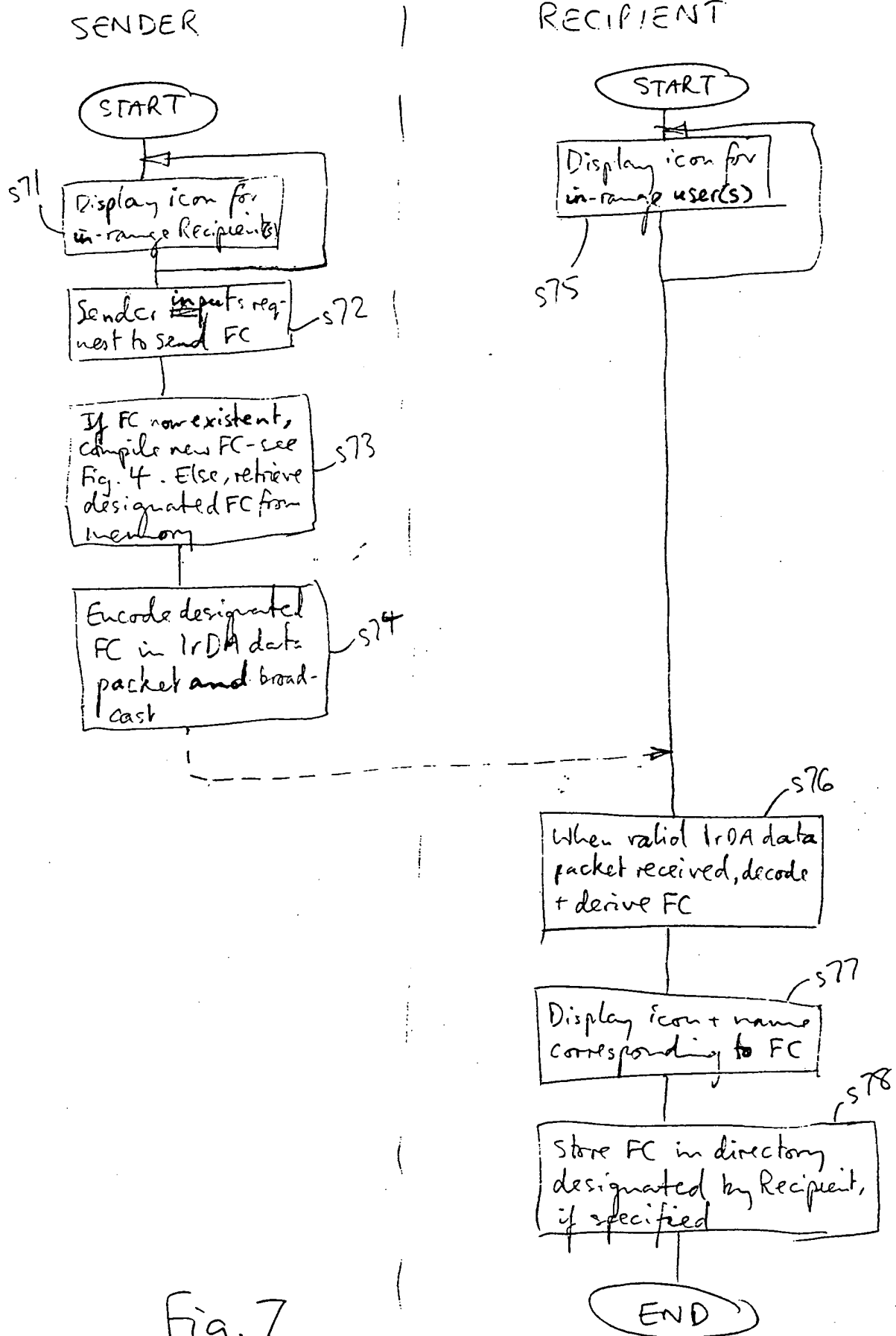


Fig. 7

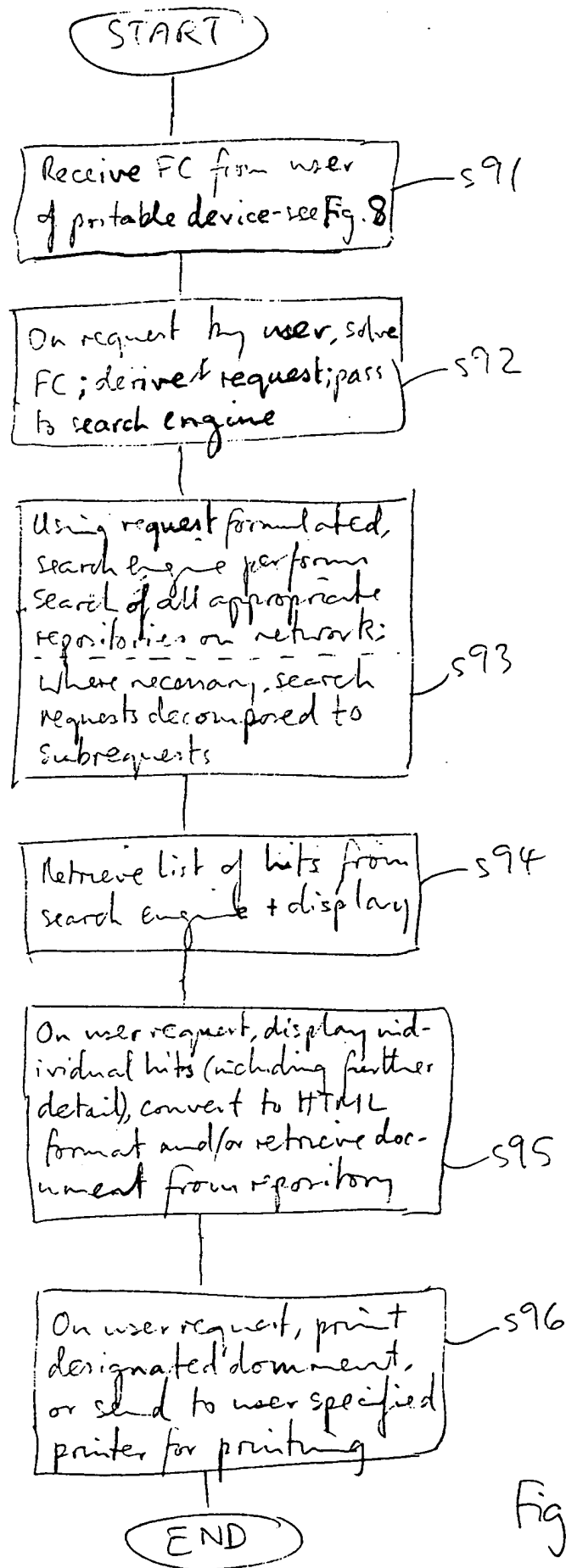


Fig. 9

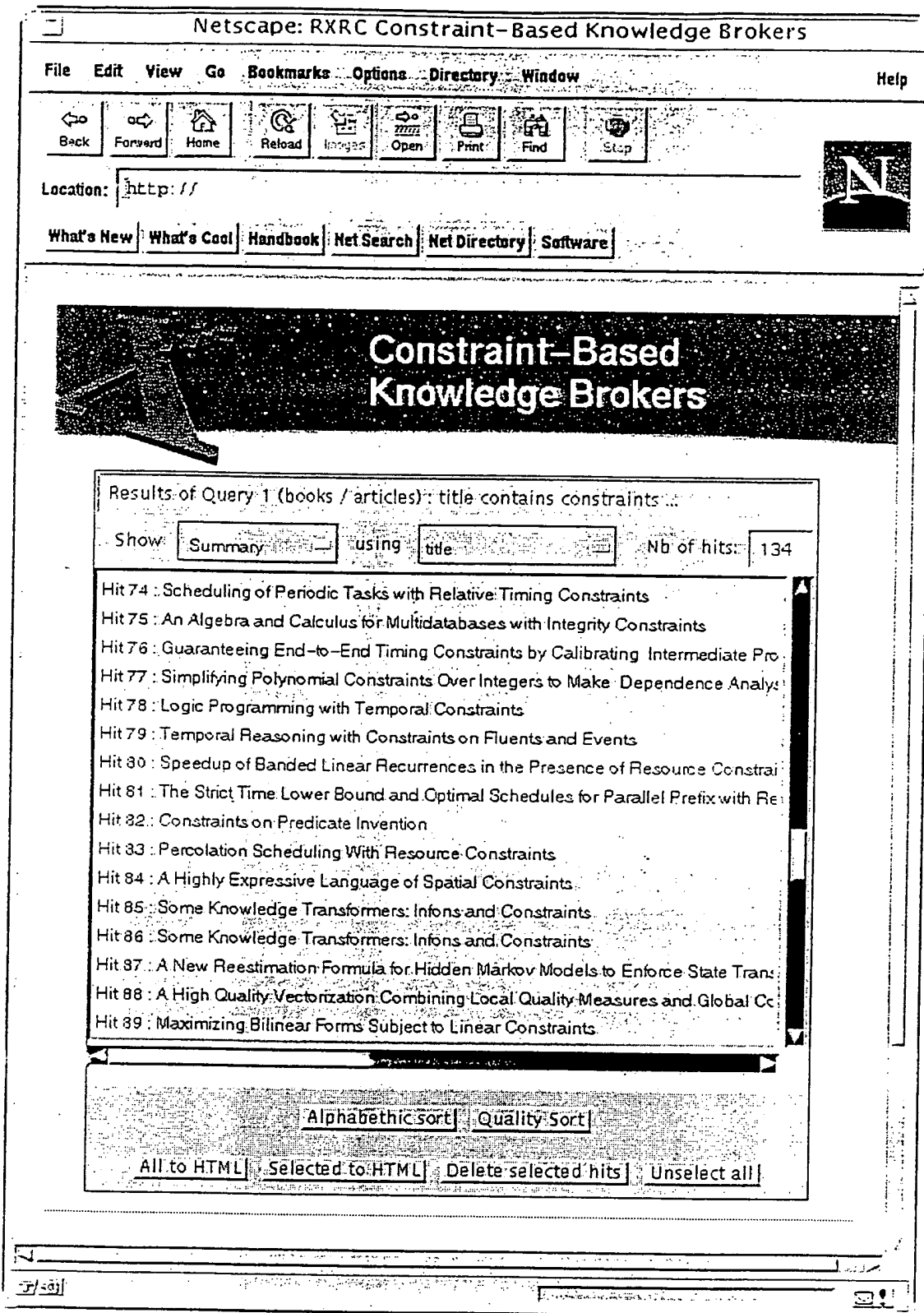


Fig. 10

6/11

Netscape: RXRC Constraint-Based Knowledge Brokers - Report

File Edit View Go Bookmarks Options Directory Window Help

Back Forward Home Reload Images Open Print Find Stop

Location: <http://>

What's New What's Cool Handbook Net Search Net Directory Software

Constraint Based Knowledge Brokers Report

Information retrieved Wed, 4 Sep 96

Query: find all books where title contains constraint and date after 90 and title contains not internet

Reference	Hit-0001
Query	Search in books servers
author	Z. Zhang.
http_url	http://www.cs.umass.edu/80/Dienad/UIL/2.0/Describe/mostrl.umass.cs%2fUM-
information_source	NCSTRL (Cornell Computer Science)
reference	UM-CS-1996-008
title	3D Reconstruction under Varying Constraints on Camera Geometry for Roboti

Reference	Hit-0002
Query	Search in books servers
author	Masato Ishizaki
bibtype	InProceedings
crossref	coling-90
information_source	Glimpse server (Arizona)

F-9 Document: Done.

Fig. 11

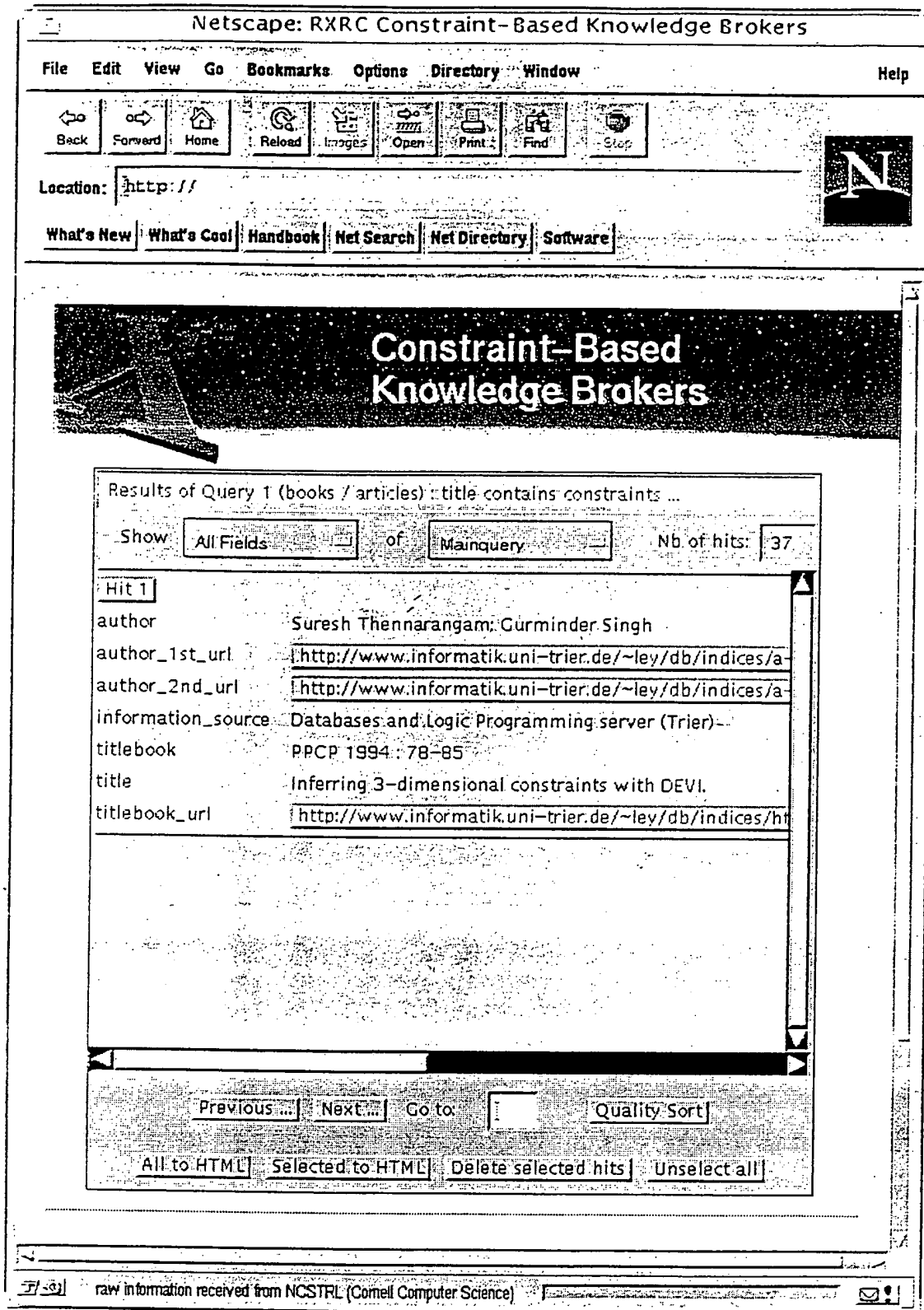


Fig.12